Amendments to Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended) A processor comprising:

instructions with at least one operand field, the instructions comprises immediate instructions having immediate data in the operand field;

a data table for storing immediate data ~~of immediate instructions~~, wherein immediate data are stored in the data table in an order determined by a flow analysis, the data table enables immediate data to be separated from the instruction stream;

a program counter for storing an instruction address of an instruction, wherein the processor fetches ~~the~~ an instruction from the instruction address in the program counter during program execution; and

an instruction decoder for decoding the instruction fetched by the processor, wherein ~~an~~ immediate data from the data table is provided to the processor if the instruction fetched is an immediate instruction.

2. (original) The processor of claim 1 further comprises an instruction register coupled to the instruction decoder, the instruction register stores the instruction fetched by the processor during program execution and passes the instruction to the instruction decoder for decoding.

3. (original) The processor of claim 2 wherein the instruction register is coupled to the data table, the instruction register provides an address of the immediate data in the data table when the immediate instruction is decoded by the instruction decoder.

4. (currently amended) The processor of claim 2 further comprises a data table addressing unit coupled to the instruction decoder and the data table, the data table addressing unit providing an address of the immediate data in the data table.

5. (original) The processor of claim 4 wherein the data table addressing unit comprises a data table pointer for storing the address.

6. (currently amended) The processor of claim 4 wherein the instruction register is coupled to the data table addressing unit, the instruction register passes addressing information to the data table ~~instruction~~ addressing unit to provide the address when the immediate instruction is decoded by the instruction decoder.

7. (original) The processor of claim 6 wherein the data table addressing unit comprises a data table pointer for storing the addressing information which serves as the address.

8. (currently amended) The processor of claim 1 further comprises a data table addressing unit coupled to the instruction decoder and the data table, the data addressing unit providing an address of the immediate data in the data table.

9. (original) The processor of claim 8 wherein the data table addressing unit comprises a data table pointer for storing the address.

10. (original) The processor of claim 8 further comprises an instruction register coupled to the instruction decoder, the instruction register stores the instruction fetched by the processor during program execution and passes the instruction to the instruction decoder for decoding.

11. (currently amended) The processor of claim 10 wherein the instruction register is coupled to the data table addressing unit, the instruction register passes addressing information to the data table ~~instruction~~ addressing unit to provide the address when the immediate instruction is decoded by the instruction decoder.

12 (original) The processor of claim 11 wherein the data table addressing unit comprises a data table pointer for storing the addressing information which serves as the address.

13. (original) The processor of claim 1 further comprises a data table addressing unit coupled to the instruction decoder and the data table, the data addressing unit provides an address of the immediate data in the data table when the instruction decoder decodes the immediate instruction.

14. (original) The processor of claim 13 wherein the data table addressing unit comprises an incrementor, the incrementor increments the address after the immediate data is provided to produce a new address for a new immediate instruction.

15. (original) The processor of claim 13 wherein the data table addressing unit comprises:

a data table pointer for storing the address; and

an incrementor, the incrementor increments the address in the data table pointer after the immediate data is provided to produce a new address in the data table pointer for a next immediate instruction executed by the processor.

16. (currently amended) The processor of claim 15 wherein the incrementor comprises an adder, the adder is coupled to the data table pointer, the adder adds the address in the table pointer and an index to produce the new address.

17. (original) The processor of claim 16 wherein the index comprises a 1.

18. (currently amended) The processor of claim 13 wherein an addressing information is passed to the data table addressing unit, the addressing information comprises an index for indexing the address to produce a new address of another immediate data in the data table for another immediate instruction.

19. (currently amended) The processor of claim 13 wherein addressing information is passed to the data table addressing unit, the addressing information comprises an index for indexing the address to produce a new address pointing to a next immediate data in the data table for a next immediate instruction fetched by the processor.

20. (currently amended) The processor of claim 18 ~~19~~ further comprises an instruction register coupled to the instruction decoder and the data table addressing unit, the instruction register stores the instruction fetched by the processor during program execution and passes the instruction to the instruction decoder for decoding, when the decoder decodes the immediate instruction, the instruction register passes the addressing information contained in the instruction to the data table addressing unit.

21. (original) The processor of claim 20 wherein the data table addressing unit comprises:

a data table pointer for storing the address; and

an adder for adding the addressing information to the address after the immediate data is provided to produce a new address in the data table pointer for a next immediate instruction executed by the processor.

22. (currently amended) The processor of claim 1 further comprises:

an instruction register coupled to the instruction decoder, the instruction register stores the instruction fetched by the processor during program execution and passes the instruction to the instruction decoder for decoding; and

a data table addressing unit coupled to the data table and the instruction register, the instruction register passing relative addressing information contained in the instruction to the data table addressing unit when the decoder decodes the immediate instruction, the relative addressing information, which comprises an index and a format indicator, is used to provide an address of the immediate data in the data table.

23. (original) The processor of claim 22 wherein the data addressing unit comprises a data table pointer containing a value,

if the format indicator comprises a post-format, the value serves as the address and after the immediate data is provided to the processor, the index is added to the value to produce a new value in the data table pointer for a next immediate instruction, and

if the format indicator comprises a pre-format, the index is added to the value to produce the address to the immediate data and the address is incremented by 1 after the immediate data is provided to the processor to produce a new value in the data table pointer for the next immediate instruction.

24. (currently amended) The processor of claim 23 wherein the format indicator comprises a binary bit having a logic 1 and logic 0 value, the logic 1 indicating the a pre-format and the logic 0 indicating the post-format.

25. (currently amended) The processor of claim 24 wherein the data table addressing unit comprises:

a first adder comprising a first input coupled to the instruction register for receiving the index ~~offset~~ and a second input coupled to an output of the data table pointer for receiving the value contained therein;

a second adder comprising a first input coupled to the instruction register for receiving the format indicator, a second input coupled to an output of the first adder, and an output coupled to the data table pointer; and

a multiplexor comprising a first input coupled to an output of the data table pointer, a second input coupled to the output of the first adder, and a select input coupled to the instruction register for receiving the format indicator to select an multiplexor output from the first and second multiplexor inputs.

26. (currently amended) The processor of claim 1 wherein the flow analysis comprises ~~data table comprises immediate data stored in an order determined by~~ a static flow analysis, the static flow analysis to ~~identify~~ identifies immediate instructions within a program.

27. (original) The processor of claim 26 wherein an immediate instruction comprises addressing information to enable the processor to retrieve a corresponding immediate data to the immediate instruction.

28. (original) The processor of claim 27 wherein the addressing information comprises absolute addressing information.

29. (original) The processor of claim 27 wherein the addressing information comprises relative addressing information.

30. (currently amended) The processor of claim 29 further comprises data table addressing unit coupled to the data table, the data table addressing unit receives the relative addressing information and produces an address ~~to the~~ corresponding to the immediate data in the data table.

31. (original) A method of executing instructions in a processor~~, the method comprises~~ by separating immediate data of immediate instructions from an instruction stream comprising:

performing a flow analysis on a program having instructions which includes immediate instructions having immediate data;

identifying immediate data from the program; and

storing the immediate data in a data table in an order determined by the flow analysis.

32. (currently amended) The method recited in claim 31 further ~~wherein separating the immediate data from the instruction stream~~ comprises:

fetching an instruction from a program;

decoding the instruction to determine a type of instruction; and

if the instruction comprises an immediate instruction, fetching an immediate data corresponding to the immediate instruction from a data table.

33. (currently amended) The method recited in claim 32 wherein the immediate data of the immediate instructions are stored in the data table in an order determined by the ~~a~~ flow analysis for identifying the immediate instructions in a program.

34. (original) The method of claim 32 or 33 further comprises providing addressing information for fetching the immediate data from the data table.

35. (original) The method of claim 34 wherein the immediate instruction provides the addressing information.

36. (original) The method of claim 35 further comprises storing the addressing information in a data table pointer.

37. (original) The method of claim 35 wherein the addressing information comprises relative addressing information.

38. (original) The method of claim 37 further comprises processing the relative addressing information by a data table addressing unit to generate an address for fetching the immediate data from the data table.

39. (currently amended) The method of claim 38 wherein processing the relative addressing information comprises:

using a value stored in a data pointer of the data table addressing unit to serve as the address; and

after fetching the immediate data, adding the relative addressing information to the value to produce a next address in the data table for fetching a next immediate data for a next immediate instruction.

40. (original) The method of claim 38 wherein the addressing information comprises an index and a format indicator.

41. (original) The method of claim 40 wherein processing the relative addressing information comprises:

if the format indicate comprises a post-format,

using a value stored in a data pointer of the data addressing unit to serve as the address, and

after fetching the immediate data, adding the index to the address to produce a next address in the data table for fetching a next immediate data for a next immediate instruction; and

if the format indicator comprises a pre-format,

adding the index to the value to serve as the address for fetching the immediate data in the data table, and

incrementing the address after the immediate data is fetched to produce a next address in the data table for fetching a next immediate data for a next immediate instruction.

42. (original) The method of claim 34 wherein the addressing information comprises relative addressing information.

43. (original) The method of claim 42 wherein providing the relative addressing information comprises:

using a value in a data pointer to provide the addressing information to fetch the immediate data from the data table; and

incrementing the value to provide a new addressing information for fetching another immediate data for a subsequent immediate instruction fetched by the processor.

44-74. (cancelled)

75. (new) A processor comprising:

a data table for storing immediate data of immediate instructions;

a program counter for storing an instruction address of an instruction, wherein the processor fetches the instruction from the instruction address in the program counter during program execution; and

an instruction decoder for decoding the instruction fetched by the processor, wherein immediate data from the data table is provided to the processor if the instruction is an immediate instruction;

an instruction register coupled to the instruction decoder, the instruction register stores the instruction fetched by the processor during program execution and passes the instruction to the instruction decoder for decoding;

a data table addressing unit comprising a data table pointer containing a value, the data table is coupled to the data table and instruction register, the instruction register passing relative addressing information contained in the instruction to the data table addressing unit when the decoder decodes the immediate instruction, the relative addressing information, which comprises an index and a format indicator, is used to provide an address of the immediate data in the data table, wherein the format indicator comprises a binary bit having a logic 1 and logic 0 value, the logic 1 indicating one of a pre-format or a post-format and the logic 0 indicating other of the pre-format or post-format;

if the format indicator indicates the post-format, the value serves as the address and after the immediate data is provided to the processor, the index is added to the value to produce a new value in the data table pointer for a next immediate instruction and if the format indicator indicates the pre-format, the index is added to the value to produce the address to the immediate data and the address is incremented by 1 after the immediate data is provided to the processor to produce a new value in the data table pointer for the next immediate instruction; and

wherein the addressing unit comprises

a first adder comprising a first input coupled to the instruction register for receiving the index and a second input coupled to an output of the data table pointer for receiving the value contained therein,

a second adder comprising a first input coupled to the instruction register for receiving the format indicator, a second input coupled to an output of the first adder, and an output coupled to the data table pointer, and

a multiplexor comprising a first input coupled to an output of the data table pointer, a second input coupled to the output of the first adder, and a select input coupled to the instruction register for receiving the format indicator to select an multiplexor output from the first and second multiplexor inputs.

76. (new) A processor comprising:

a data table for storing immediate data of immediate instructions;

a program counter for storing an instruction address of an instruction, wherein the processor fetches the instruction from the instruction address in the program counter during program execution; and

an instruction decoder for decoding the instruction fetched by the processor, wherein immediate data from the data table is provided to the processor if the instruction is an immediate instruction;

an instruction register coupled to the instruction decoder, the instruction register stores the instruction fetched by the processor during program execution and passes the instruction to the instruction decoder for decoding;

a data table addressing unit comprising a data table pointer having a value, the data table addressing unit receives relative addressing information for decoded immediate instructions, the relative addressing information, which comprises an index and a format indicator, is used to provide an address of the immediate data in the data table;

if the format indicator indicates a post-format, the value serves as the address and after the immediate data is provided to the processor, the index is added to the value to produce a new value in the data table pointer for a next immediate instruction and if the format indicator indicates a pre-

format, the index is added to the value to produce the address to the immediate data and the address is incremented by 1 after the immediate data is provided to the processor to produce a new value in the data table pointer for the next immediate instruction; and

wherein the addressing unit comprises

a first adder comprising a first input coupled to the instruction register for receiving the offset and a second input coupled to an output of the data table pointer for receiving the value contained therein,

a second adder comprising a first input coupled to the instruction register for receiving the format indicator, a second input coupled to an output of the first adder, and an output coupled to the data table pointer, and

a multiplexor comprising a first input coupled to an output of the data table pointer, a second input coupled to the output of the first adder, and a select input coupled to the instruction register for receiving the format indicator to select an multiplexor output from the first and second multiplexor inputs.